Reply Brief

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BOARD OF PATENT APPEALS AND INTERFERENCES

In re patent application of:
  Glider et al.

Serial No.: 10/723,085

Filed: November 26, 2003

Group Art Unit: 2192

Examiner: Zheng Wei

Atty. Docket No.: ARC920030081US1

For: SOFTWARE UPGRADE AND DOWNGRADE IN SYSTEMS WITH
PERSISTENT DATA

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPELLANTS' REPLY BRIEF
Sirs:

This Reply Brief is being filed in response to the Examiner's Answer dated
October 29, 2008, is being filed within two months of the date of the Examiner's Answer
and is therefore timely filed.

Generally the claimed invention involves a method of upgrading and
downgrading software maintained by nodes within a computer network. As explained in
paragraph 3 of Appellants' specification, it is often necessary to upgrade or downgrade
software maintained by nodes within a computer network to allow proper communication
between the nodes.

Independent claims 1, 7, 13, and 15 recite in relevant part,

"...applying an upgrade to a first next level of software ... ;

...applying an upgrade to a second next level of software ... ;

...applying a downgrade to a first previous level of software ... ;

...applying a downgrade to a second previous level of software ... ."

Between the Various "applying" steps, the method converts persistent data structures. Therefore, the claimed invention performs upgrades and downgrades on two different "next" levels of software and two different "previous" levels of software.

The rejection relies upon U.S. Patent Application Publication No. 2003/0092438 to Moore et al., hereinafter, Moore, to teach the ability to upgrade or downgrade an application and to convert data to a new version format. For example, this is shown in Figure 4 items 118 and 120 of Moore.

However, it is admitted that Moore does not explicitly disclose two-level software upgrading (Examiner's Answer, page 4, second paragraph). Therefore, the Office has applied U.S. Patent No. 6,385,770 to Sinander et al., hereinafter, Sinander, for teaching two-level software upgrading. For example, as shown in Figure 1 of Sinander, software can be upgraded from version V_0 to version V_1; and then upgraded from version V_1 to version V_2. The Office proposes that this aspect of Sinander teaches the claimed invention and Appellants disagree. This disagreement is central to the present rejection, and is the primary focus of this Reply Brief. The tertiary reference (Schroder) is applied merely to demonstrate that software can be upgraded without traffic interruption.

In the paragraph bridging pages 7 and 8 of the Examiner's Answer, the Office explains that the first and second parts of the upgrade framework of Sinander are being used to upgrade content 1 and content 2, wherein each of the upgrade contents comprises tasks specific for the corresponding software system upgrade. Thus, the upgrade is from software system version V0 to software system version V2. The Examiner's Answer further explains that when the upgrade framework is executed or applied, the upgrade content 1 will be carried out, upgrading the system from software system version V0 to V1. Thereafter, upgrade content 2 is executed, upgrading the software system from V1 to version V2. Therefore, the Examiner's Answer states that Sinander discloses that after the specific task of upgrade content 1 are completed, upgrade content 2 is executed. This

involves specific tasks, now specific for upgrade content 2. The Examiner's Answer argues that the upgrade content 2 task of a second part of the upgrade framework also includes resuming operations, now in accordance with the new software system version. Thus, the Examiner's Answer argues that the "upgrades" of Sinander are applied to multiple software version levels, i.e. from V0$\rightarrow$ V1 which is the first level of software and from V1$\rightarrow$ V2 which is corresponding to the second next level of software as the Appellants argued.

However, as can be most clearly seen by the paragraph appearing in column 3, lines 54-58 of Sinander, the upgrades are sequential upgrades and can be considered as two different steps within a single upgrade. To the contrary, the claimed invention provides different upgrades and different downgrades with intervening conversions of the persistent data structures depending upon what each specific node requires. Therefore, while Sinander essentially discloses an upgrate that is performed in two steps, the claimed invention describes four separate processes.

If the reader would indulge the undersigned in some necessary detail to highlight this distinction, reference is made to Appellants' Figures 6(a) and 6(b) that illustrate the claimed software upgrade technique. Details can also be found in paragraphs 32-34 of Appellants' specification. In this example, it is assumed that there are three nodes 50a, 50b, 50c, represented as circles in the various figures, with the distributed application using the existing (old) software version X. During the software revision process, it is determined that an update to the software application is needed such that the persistent structure with format D must be modified resulting in new data structure format D*. The software upgrade is applied on each node 50a, 50b, 50c sequentially one node at a time, until all of the nodes 50a, 50b, 50c in the system 500 have been upgraded.

The software upgrade level X+1 is first applied on a single node 50a as shown in Figure 6(a). Here, persistent data that had been stored by software level X is retrieved. According to this example, the persistent data in version X is indicated as format D. The software level X+l understands both data formats (D from existing version and D* in the new version) but in a conversion step converts all persistent data from the D* format to

the D format. The data structures can now be used as the rest of the software at level X+1 expects format D. Thus, the persistent data will be stored in representation format D. Next, for both the other nodes 50b, 50c in the distributed system 500, the same steps described above are repeated one node at a time, to upgrade the entire system 500 to software level X+1.

Thereafter, there will be another level that uses format D* such that the software code expects format D* since during the software revision process it is determined that an update to the software is needed such that the persistent structure D located at each node 50a, 50b, 50c is modified resulting in persistent data structure D*. Then, as shown in Figure 6(b), the invention applies a software upgrade to level X+2 on a single node 50a. Here, the persistent data that had been stored by software level X+1 is retrieved. As indicated above, this persistent data will be in format D. As software level X+2 understands both data formats (D and D*), it converts all persistent data from format D to format D*. The data structures can now be used as the rest of the software at level X+2 expects D*. Persistent data will be stored in representation format D*. Thereafter, for both the other nodes 50b, 50c in the distributed system 500, the above steps are repeated one node at a time to upgrade the overall system 500 to software level X+2. Figure 6(c) and 6(d) illustrate the similar software downgrade technique.

Thus, Appellants submit that, in Sinander, the upgrades are sequential upgrades and can be considered as two different steps within a single upgrade. To the contrary, the claimed invention provides different upgrades and different downgrades with intervening conversions of the persistent data structures. Therefore, Appellants submit that the prior art of record does not teach or suggest" applying an upgrade to a first next level of software … ; … applying an upgrade to a second next level of software … ; … applying a downgrade to a first previous level of software … ; … applying a downgrade to a second previous level of software … .", with intervening conversions of persistent data structures as recited in independent claims 1, 7, 13, and 15.

Reply Brief

      Therefore, Appellants respectfully request that the Board reconsider and withdraw the rejections of all of the pending claims, i.e., claims 1-5, 7-11, 13, and 15-19 and pass these claims to issue.

      Please charge any deficiencies and credit any overpayments to Attorney's Deposit Account Number 09-0441.

Respectfully submitted,

Date: December 24, 2008

/Frederick W. Gibb, III/
Frederick W. Gibb, III

Registration No. 37,629
Gibb IP Law
2568-A Riva Road, Suite 304
Annapolis, MD, 21401
Voice: (410) 573-1545
Fax: (301) 261-8825
Customer No. 29154